[HoSt02] *Matthias Hollick, Ralf Steinmetz;* **Security for Ad-Hoc Service Information - Threat Analysis of the Service Location Protocol;** Enterprise Security, it Verlag für Informationstechnik GmbH, Höhenkirchen, March 2002, S. 163-176.

Security for Ad-Hoc Service Information

Threat Analysis of the Service Location Protocol

Matthias Hollick[1], Ralf Steinmetz[2]

[1]Fraunhofer IPSI, Darmstadt
Bereich Mobile Interaktive Medien
matthias.hollick@ipsi.fraunhofer.de

[2]Technische Universität Darmstadt
Fachbereich Elektrotechnik und Informationstechnik
Lehrstuhl für Multimedia Kommunikation
ralf.steinmetz@kom.tu-darmstadt.de

Abstract

The discovery of appropriate information about networked services is a fundamental requirement to enable networking, ranging from small-scale ad-hoc to enterprise-scale networks. The IETF's Service Location Protocol (SLP) allows for lightweight service discovery, targeted towards zero-configuration networking in static and ad-hoc systems which base on Internet technology. Service discovery protocols for ad-hoc usage or introducing dynamic behaviour in large scale systems often declare security optional or do not care for it at all, also we assume dealing with security crucial. This contribution clarifies upon the threats to service information in case of SLP. After a brief description of the protocol mechanisms we present an in depth threat analysis of the Service Location Protocol with respect to service information security from a systems perspective. The possible usage scenarios and corresponding attacks are described and subsequently visualized using attack trees. We conclude with outlining some of the open security issues within SLP.

Keywords

Network Security, Service Location, Service Discovery, Service Location Protocol, Threat Modeling.

1 Introduction and Motivation

The advent of networked services enriched the pure networking infrastructure they built upon and laid the foundation for the broad success of the Internet technology in all its facets. With the paradigms of networking shifting towards user-, system- and service-mobility the infrastructures build for static networking become increasingly insufficient. To support current and future usage paradigms like nomadic, pervasive or ubiquitous computing combined with the dynamic creation and usage of services within enterprise networks [Weis91][Klei00], a change towards ad-hoc supporting seamless assistance for configuring networked services is crucial. As a starting-point service discovery and service location are the areas of interest when we seek for networked services. More precisely they refer to the provision of mechanisms to register and retrieve service information.

Starting with the real world scenario of nomadic computing [Klei00] it is likely to face the challenge described above. Assuming a nomadic enclave to host a couple of nomads, there will be the need to solve the ad-hoc micro-nomadicity problem of finding appropriate networked services within the nomadic enclave [Holl01]. To solve configuration issues in such environments there are various frameworks and protocols. The Service Location Protocol [VeGu97] [GuPe99] presents a lightweight but sufficient approach to enrich basic configuration methods like the Dynamic Host Configuration Protocol (DHCP) [Drom97] or supplement static or centralized service information directories like the Domain Name Service (DNS) with its service resource records [Mock87][GuVE00]. Hereby, the emphasis of SLP lies on supporting dynamic ad-hoc scenarios.

However, service information is worth to be considered a security critical aspect. Imagine a malicious user who wants to find attack goals. Nowadays he likely will have to monitor network traffic and hence needs the capability to eavesdrop. If service information is available for free and unprotected it may be enough for the attacker to ask for the services he is interested in and get them delivered on a silver plate. "Security by obscurity is no security" is a common understanding within the security community but with respect to service information we argue that some regulation may be vital to permit security.

This paper identifies the most important security related issues coupled to SLP by means of a detailed threat analysis. After a brief introduction to the base function and core mechanisms of SLP we distinguish the different modes of usage related to security. Following, the threat analysis from a systems perspective and appropriate visualization of the threats build the key part of this contribution. We conclude presenting the weak and missing links in SLP-Security leading over to an outlook of future research activity.

2 The Service Location Protocol

2.1 Base Protocol Function

To fulfill the need of service discovery and selection SLP provides a foundation to build upon. Starting with an IP-address, which in ad-hoc scenarios can be gained autonomously through auto-configuration for IPv4, stateless address auto-configuration for IPv6 [ThNa98] or assigned by means of external mechanisms like DHCP, a network node may start communicating - if it knows about services to use. To minimize the administrative burden and to adapt to ad-hoc situations SLP allows for services to be dynamically published and retrieved. To do so, it assigns additional roles to existing network entities for the purpose of service discovery. These roles are so called *User Agents (UA)*, *Service Agents (SA)* and *Directory Agents (DA)* and are related to application processes. The UA acts on the users behalf to request and retrieve service information from an SA or DA. The SA is representing one or more services by means of advertising them while the DA acts as service information cache, which collects service advertisements and as a result allows for better scalability. The service information is delivered using Uniform Resource Locators (URLs) [GuPK99]. Moreover, the data structure maintained by the SLP entities comprises unique service types and description attributes, which are defined within so called service templates [GuPK99]. Besides the basic functions of registering and retrieving service information SLP features scopes to allow grouping of service information by users or administration.

2.2 Direct vs. Indirect Mode of Service Discovery

Within SLP we can distinguish two usage scenarios. One covers the direct information exchange between UA and SA, which we will refer to as the *direct mode* feasible for small networks. The second uses DAs providing cached information and therefore will be denoted *indirect mode.* Moreover, the protocol design can be divided into a *control-plane* and a *data-plane.* The control plane is used to find nearby agents for entering of or to apply for service information. The data-plane thereafter deals with exchanging the very service information. When using SLP in direct mode control- and data-plane cannot be separated clearly. Table 1 depicts the message types within SLP, their function is described below.

Tab. 1. Supported Message Types in SLPv2

Mandatory Message Types		*Optional Message Types*	
Message Type	**Abbreviation**	**Message Type**	**Abbreviation**
Service Request	*SrvRqst*	Service Deregister	*SrvDeReg*
Service Reply	*SrvRply*	Attribute Request	*AttrRqst*
Service Registration	*SrvReg*	Attribute Reply	*AttrRply*
Service Acknowledge	*SrvAck*	Service Type Request	*SrvTypeRqst*
DA Advertisement	*DAAdvert*	Service Type Reply	*SrvTypeRply*
SA Advertisement	*SAAdvert*		

Interaction in direct mode is started on behalf of the UA, which issues a multicast *SrvRqst* including the characteristics of the service in question. SAs with appropriate information will return an unicast *SrvRply* with the correspondent Service URL (see fig. 1). Unsolicited *SAAdvert* messages from the SA can be described as a control-plane message to periodically inform UAs of their existence.

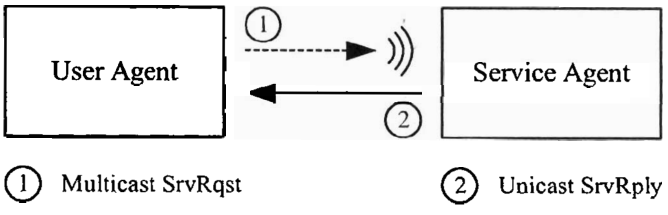

Fig. 1. SLP protocol exchange for direct mode

The control-plane of the indirect mode manages the establishment of the communication-association between UA and DA and consists of an active discovery of the DA through multicast *SrvRqst* messages from the UA (asking for directory service services) which are answered by a unicast *DAAdvert* reply. Complementing the DA may be multicasting *DAAdvert* messages (passive discovery) (see fig. 2).

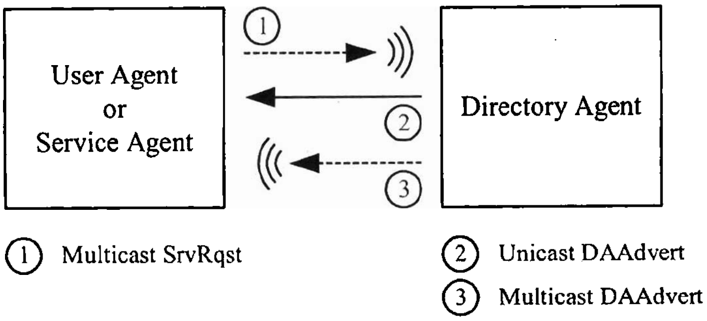

Fig. 2. Control plane for indirect mode

After completion of the control-plane the data-plane in indirect mode is made of the registration sequence initiated by the SA with a unicast *SrvReg* and the corresponding unicast *SrvAck* of the DA. The UA - DA interaction equals direct mode but uses the respective unicast messages *SrvRqst* and *SrvRply* (see fig. 3). The optionally to implement request messages (*AttrRqst*, *SrvTypeRqst*) work multicast based in direct mode, unicast in indirect mode. The optional reply messages (*AttrRply*, *SrvTypeRply*) and the *SrvDeReg* use unicast transport.

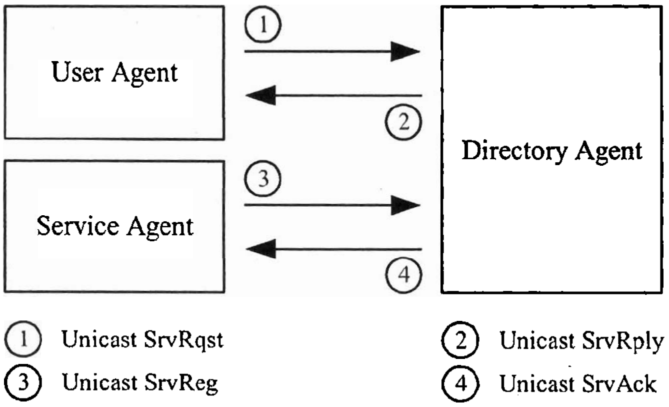

Fig. 3. Data plane for indirect mode

2.3 Security within SLP

Having explained the function of the Service Location Protocol we will peek into the security mechanisms in place. Since SLP was designed as a lightweight protocol there are only few security features in place. Moreover, these are optional.

Security supported within SLP includes authentication within one administrative domain if preconfigured security-associations exist among the hosts. Authentication blocks can be requested from UA or DA to ensure the integrity of the service information by means of authenticating the DA or SA. SLP has no other security mechanisms in place and thus confidentiality or access-control are not supported by the protocol.

For further evaluation of security we will use the above made distinction between use modes. The direct communication between UA and SA which is likely to happen in small and unorganized ad-hoc scenarios and the indirect communication likely in medium to large and administratively organized ad-hoc scenarios (e.g. a nomadic enclave provided for guest access to network resources will likely consist of a DA to better enforce policies). Following we will investigate both usage paradigms in respect with security threats possibly mounted against the protocol, the implementation or the entities.

3 Threat Analysis and Visualization

To visualize the threats we use attack-trees likewise in [Schn00][Ecke00]. Attack trees are a methodical way of describing possible threats against systems and thus facilitate developing countermeasures against these threats. Moreover, there is the possibility to extend the meaning of the trees by attributing the possible attacks with e.g. cost of the attack.

Attack trees represent a snapshot of all known attacks, attacks the tree-builder was not aware of or exceed his assumed threat-model may be lacking in the tree. However even with some threats missing we consider attack trees an excellent mechanism for starting a risk analysis. For a better understanding fig. 4 depicts an example attack tree for a classical threat against a physical safe [Schn00].

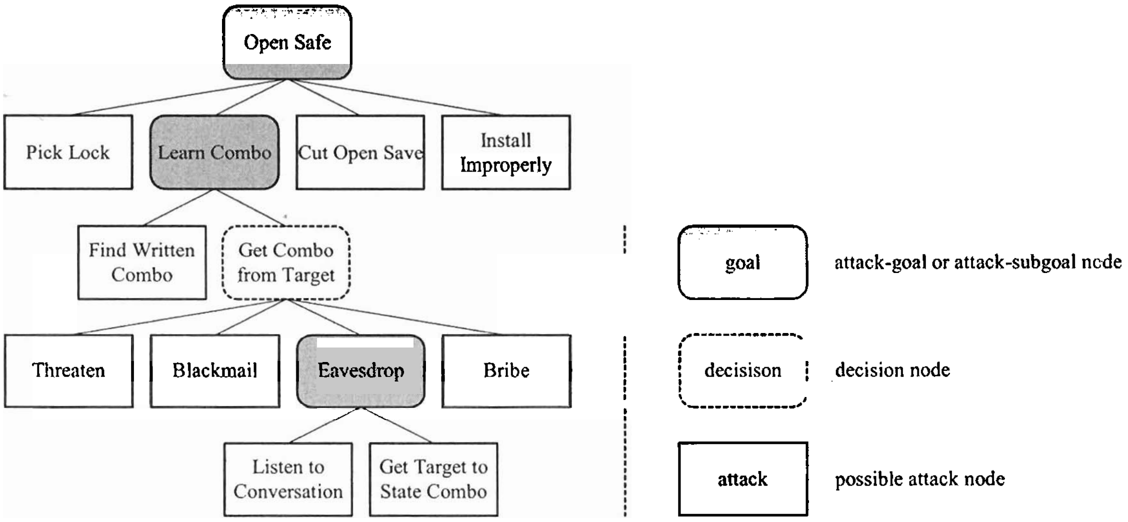

Fig. 4. Example attack tree and legend

4 The Threats

If we dig into threat modeling, the capabilities of the adversaries have to be specified in advance. We expect a powerful attacker with ability to eavesdrop and manipulate network traffic in flight. Our attacker can tamper the software of all entities involved and has physical access to machines. Because of this broad capabilities there surely will some attacks be missing in the following but dealing with system security always includes not looking for the perfect but the best solution reachable.

We regard counterfeiting the service information (*SrvInfo*) as a primary attack goal and thus as the root of the attack tree. To better illustrate the impact of tampering with service information the following possible results may be achieved: Assume the correct Service URLs are *A:\\locationA* for service-A (the Service URL is simplified for a better understanding) and *B:\\locationB* for service-B. For *SrvRqst* messages asking for both service-types, the possible answers of a SLP entity (under attack) can be classified as shown in table 2 and depicted in figure 5. Moreover, there are timing constraints only implicit in table 2. Since service deregistration is not mandatory, the service information will time-out. E.g. a correctly delivered answer from a DA may tell about services that are no longer available or have meanwhile been spoofed. This may result in redirecting clients effectively. As a conclusion the validity of service information includes variables outside of the SLP framework.

Tab. 2. Attack Results Against Service Information

No.	Result	Resulting URL
1	No *SrvRply* is available	*No SrvRply available*
2	*SrvRply* is avail.	*(Metagroup)*
2.1	*SrvRply* is avail. but empty	*No URL available*
2.2	*SrvRply* is avail. and complete	*A:\\locationA, B:\\locationB*
2.3	*SrvRply* is avail. and incomplete	*A:\\locationA*
2.4	*SrvRply* is avail. but *SrvInfo* is Inconsistent	*(Metagroup)*
2.4.1	*SrvRply* contains the complete plus manipulated *SrvInfo*	*A:\\locationA, B:\\locationB, X:\\locationX*
2.4.2	*SrvRply* contains incomplete plus additional manipulated *SrvInfo*	*A:\\locationU, B:\\locationB*
2.4.3a	*SrvRply* contains only manipulated *SrvInfo*	*A:\\locationU, B:\\locationV*
2.4.3b		*A:\\locationU, Y:\\locationY*

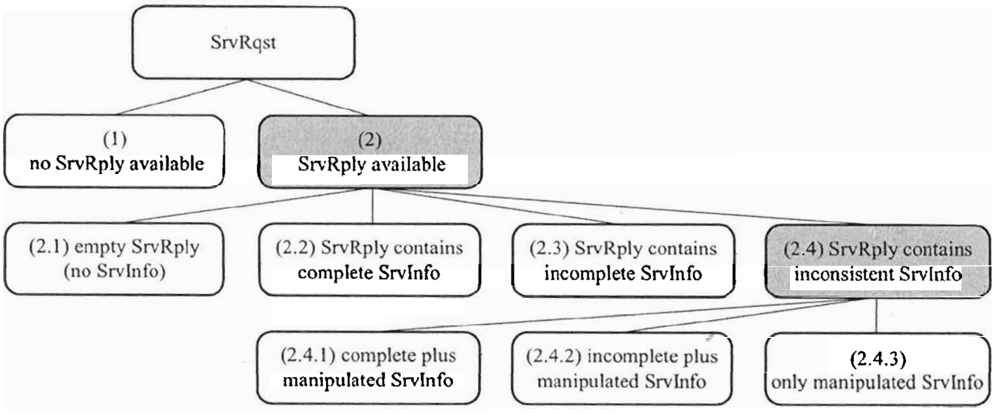
SrvRqst
(1) no SrvRply available
(2) SrvRply available
(2.1) empty SrvRply (no SrvInfo)
(2.2) SrvRply contains complete SrvInfo
(2.3) SrvRply contains incomplete SrvInfo
(2.4) SrvRply contains inconsistent SrvInfo
(2.4.1) complete plus manipulated SrvInfo
(2.4.2) incomplete plus manipulated SrvInfo
(2.4.3) only manipulated SrvInfo

Fig. 5. Tree Structure of Possible Attack Results Against Service Information

As already mentioned we divide the attack trees for better visualization. For better granularity we make a further distinction with respect to the entity under attack. Thus we can build a backbone tree interconnecting the discrete attack trees (see fig. 6). We investigate all scenarios but the shaded areas in fig. 6 are not shown in individual attack trees due to space limitations and similarities to the trees already mentioned.

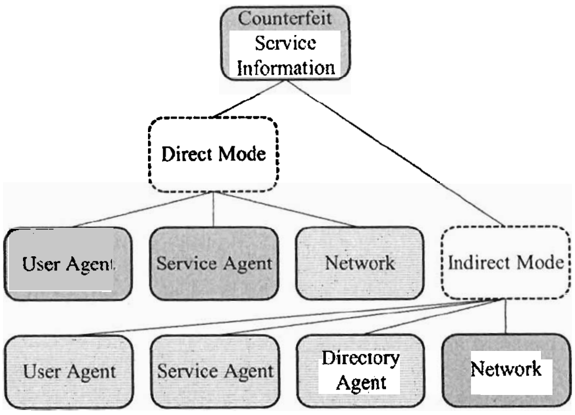

Fig. 6. Division of the attack-goal into sub-goals

Derived from the ad-hoc usage paradigm of the direct mode we regard the case that the SLP-authentication is not implemented and only the mandatory protocol features are implemented and used. Though we will mention the attacks being impossible if using integrated SLP security subsequently. The attack tree for the net will include all possible protocol interactions.

4.1 Threats in Direct Mode

We will start investigating the attacks in direct mode, be-cause of the easier protocol interactions involved. Attacks can be mounted against underlying or superseding processes on the same machine the agent resides on. The agent process itself and the network are attack goals, too. For the following we will study attacks against UA in fig. 7 and SA in fig. 8, which are also relevant in indirect mode.

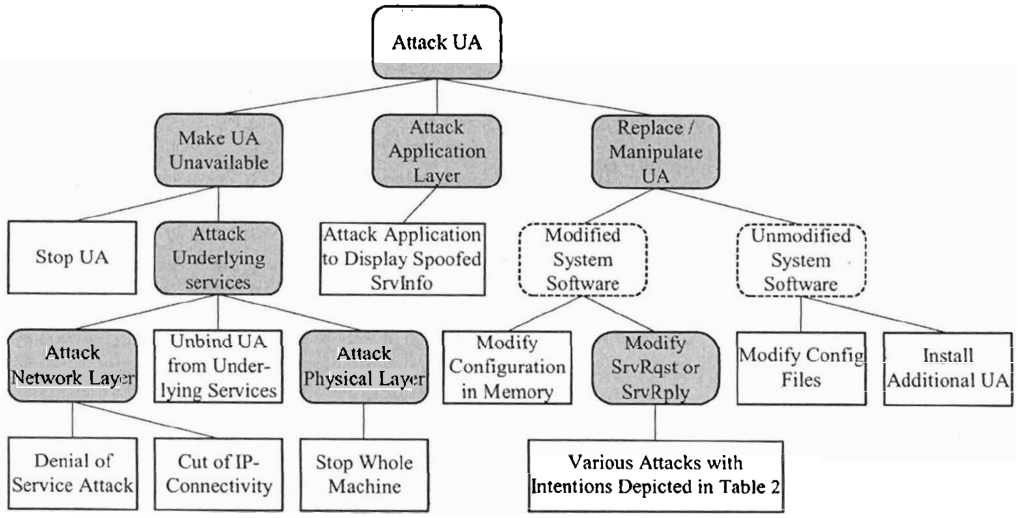

Fig. 7. Attacks against User Agents in direct mode

Possible attacks against the UA include turning off the device (if not physically protected) or stopping the UA service. The underlying network stack can be manipulated to shift the socket to a malicious UA or the application presenting the results can be tampered with.

The normal bootstrap mode of an SLP UA includes first contacting DAs and configuring the scope if provided by means of DHCP-Options [PeGu99] (spoofing DHCP-Servers is pretty easy and thus presents a major threat since in direct mode there are no DAs available by definition). Second preconfigured settings are used if available. Thus manipulation of the configuration settings presents a powerful attack. Step three includes a multicast discovery of DAs available and step four, if reached, falls back to multicast discovery of all services directly with SAs within reach. Within our ad-hoc scenario there will be no DAs available and so spoofing one is a strong attack because it will be used per protocol design.

Assuming no malicious DA in reach, the direct *SrvRqst* and *SrvRply* with SAs will happen. Hereby the UA can be tricked by false SAs or replayed *SrvRply* messages, which will overwrite the replies received before, depending on the implementation. Setting up a second UA on the same machine also may conflict with the integrity of the service information.

The SA as second entity can be attacked in various ways, too. Tampering with configuration files, introducing faked DAs, or registering of false alias-services is within reach. Each of the results in table 2 can easily be reached by tampering with the System hosting the SA. Adversaries are likely to set up bogus services registering on behalf of correct services and redirecting the traffic. Furthermore, fiddling with attributes within the registration at SA side may lead an UA to chose a "better" service and thus reduce the likelihood of choosing the original one. (E.g. imagine a system spotting to be a service version "XP" and thus blaming all the old "2000" versions out there).

The network in between SA and UA may be the most powerful attack tool if under full-control and will be investigated in indirect mode later. To summarize the threats in direct mode, we can complain that the reduced assumption of two entities (SA and UA) using two messages (*SrvRqst* and *SrvRply*) shows to be a pretty complex system and can be damaged in various ways. The system can be easily attacked e.g. by using false agents or mounting attacks against the network stack on the same machine. To tamper with the system, using faked services, doesn't need root or administrative privileges and can be done as normal user in today's implementations.

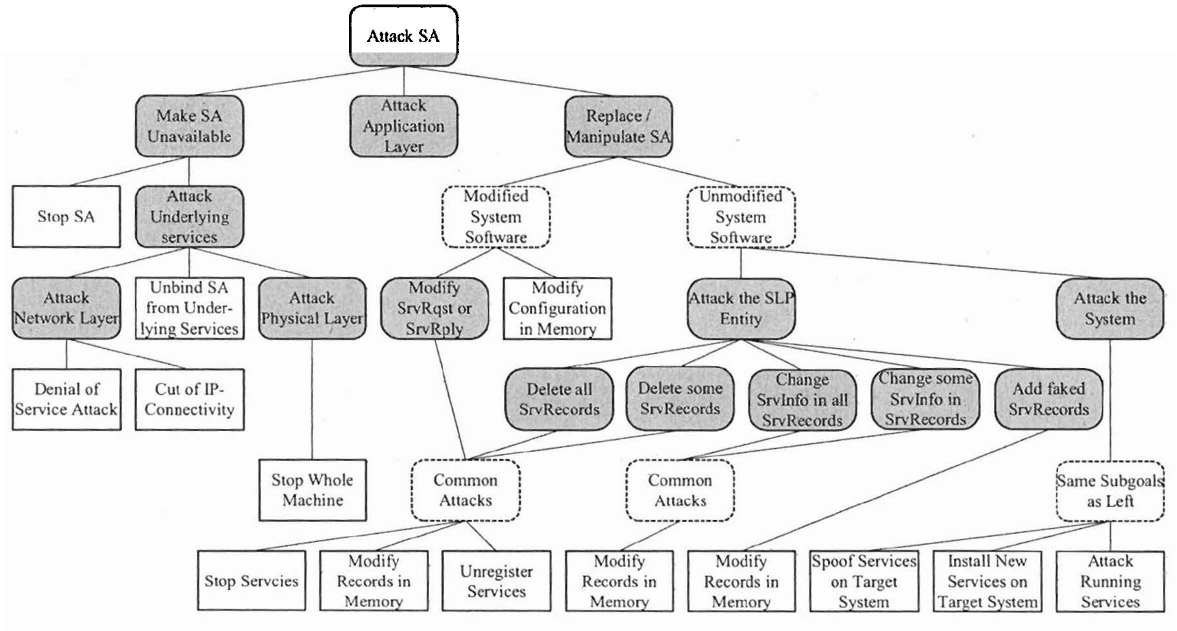

Fig. 8. Attacks against Service Agents in direct mode

4.2 Threats in Indirect Mode

The possible attacks in indirect mode are richer and even worse than in direct mode, because with the DA an additional entity is involved. Additionally, the division in control and data plane, which has richer meaning, compared to direct mode, makes the system more complex and thus more vulnerable. On the other hand, we assume indirect mode to be implemented in administrated environments. Users in a nomadic enclave conventionally will be supported by a preconfigured DHCP-server and DAs. The optional protocol features are likely to be implemented because the number of choices among services is enriched.

The possible attacks against the UA equal the attacks in direct mode and thus the attack tree is equal, too. Since we expect the configuration taking place with preconfigured DHCP scopes and DAs, attackers are likely to make the DHCP-Server or the referred DA unavailable and/or replace them (see network attack tree in fig. 9).

Attacks against the SA include the same ones as in direct mode. Since the analysis in indirect mode includes the optional messages spoofed service deregistrations have to be regarded a thread. The DA acts as a cache between UA and SA. Altering this cache is very effective - without authentication it can be realized using spoofed IP-addresses and faked registration or deregistration messages. Moreover, a simple amok-SA can mess up the availability of the DA system by registering zillions of unsolicited services. An amok-DA with eavesdropping capabilities can answer all *SrvRqst* messages and thus introduce additional (false) information. Having the ability to modify system software presents the possibility to add unwanted extensions to the reply-messages or just send away false information.

There is no particular attack tree for the attacks against DAs, because due to the caching function a DA equals one or more SAs and can be attacked the same way. [GuPe99] states that there should be only one DA registered per UA at a time. This DA turns out to be a single point of failure if replaced or attacked. A system, which consists of a proper functioning DA, can also be abused to seek for configuration information (here correctness of the information is the faulty condition). An attacker normally steps into a network peeking for worthy goals. If SLP is in place and provides a large information base, the attacker only has to issue a *SrvTypeRqst* to learn about all services registered. Subsequently issued *SrvRqst* and *AttrRqst* messages can gather all the information available.

Attacking the network (see fig. 9) shows all possibilities of tampering with native SLP messages. By modifying packets in flight all possibilities stated in table 2 can be reached. and would make it easy to mix up service information in question. Attacking network entities like DHCP servers, destroying multicast connectivity or using denial of service attacks against the SLP-daemons can render the protocol useless, too. Fig. 9 concentrates on the attacks against the network in general, but we have to keep in mind more subtle attacks like manipulating the *XID-field* or the *fresh-flag* within SLP-headers. Additionally we have to think about constructing spurious messages or denial of service attacks introducing chaos to the SLP-system.

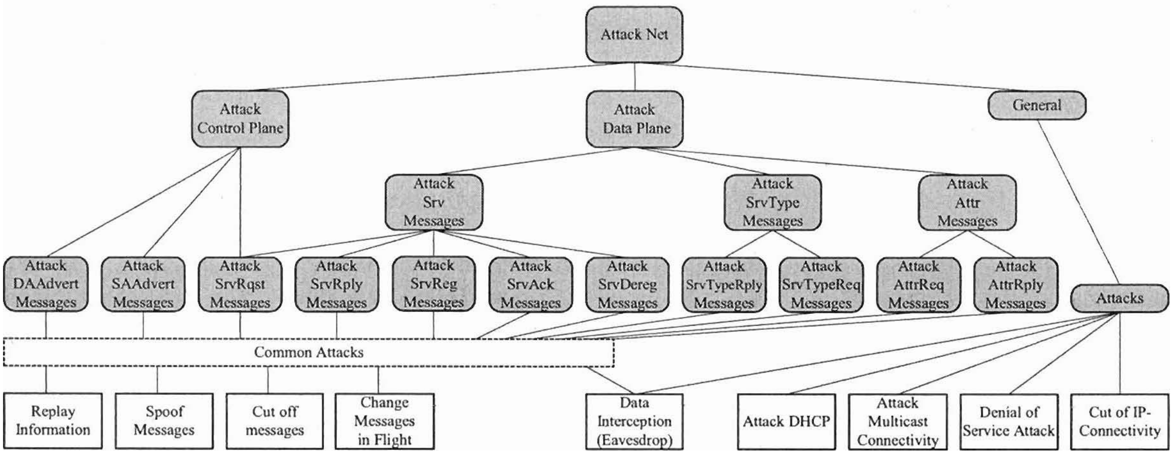

Fig. 9. Attacks against the net in indirect mode

Since SLP relies on multicast communication attacks against multicast groups have to be considered. The ability to introduce packets in the network can be used to spoof all entities taking part in the communication.

4.3 Summary of Threats

Our broad investigation of threats against the Service Location Protocol leads to a deeper understanding of risks in the area of service information. Not only the authenticity or integrity of the service information, which partly can be gained using SLP, may be considered, but also the confidentiality or privacy of service information. A distributed but partly insecure framework like SLP benefits in ad-hoc scenarios in terms of stability but may be damaged through easy to introduce entities taking part in the distributed system and confusing it.

Service information for networked services in ad-hoc environments clearly has to consider security. SLP introduces security only if preconfigured security associations exist and restricted to authentication of the communication partners.

5 Conclusion

Networked services are crucial towards actual and future usage paradigms like nomadic, seamless or ubiquitous computing. Service location is beneficial esp. for ad-hoc scenarios or highly dynamic enterprise networks to allow for easy configuration of systems and to make dynamic services viable to the user.

The service location protocol presents a flexible framework to be used to obtain service information. However the protocol design was not done with security in mind. The possible weaknesses we have documented are mostly related to protocol design and partly related to implementation issues. It is obvious that the security-aspects within frameworks and protocols for lightweight service discovery need to be emphasized for the future. The different modes of communication between SLP entities volunteer for different security mechanisms to be in place.

Today's common argumentation to protect services and give away service information for free and unprotected cannot keep up with future usage paradigms. Service information is worth being protected and even lightweight protocols like SLP should withstand basic attacks by design.

Future work will try to improve the security features of SLP by design and implementation. We are underway studying the implications of service discovery and started modelling the security needs of service information more generic. Our goal is to theoretical provide a common understanding of security in the field of service information for ad-hoc usage.

Acknowledgement

This work is partly supported by the German Research Council (DFG) as part of the program Security in Information and Communication Technologies within the SECCO project.

References

[Drom97] R. Droms, RFC 2131 - Dynamic Host Configuration Protocol, Proposed Internet Standard, March 1997.

[Ecke00] Claudia Eckert; IT-Sicherheit Konzepte - Verfahren - Protokolle, Oldenburg Verlag, 2000, ISBN: 3-486-25298-4

[GuPe99] Erik Guttman, Charles E. Perkins, and Michael Day, RFC2608 - Service Location Protocol, Version 2, Proposed Internet Standard, June 1999

[GuPK99] Erik Guttman, Charles E. Perkins, James Kempf, RFC2609 - Service Templates and Service: Schemes, Proposed Internet Standard, June 1999

[GuVE00] A. Gulbrandsen, P. Vixie, L. Esibov, RFC2782 - A DNS RR for specifying the location of services (DNS SRV), Proposed Internet Standard, February 2000

[Holl01] Matthias Hollick, Secure Service Centered Networking for Nomadic Usage, to appear in Proceedings of CMS 2001, Kluwer, May 2001

[Klei00] L. Kleinrock, On Some Principles of Nomadic Computing and Multi-Access Communications, IEEE Communications Magazine 38, p.46–50, July 2000

[Mock87] P.V. Mockapetris, RFC1035 - Domain names - implementation and specification, Proposed Internet Standard, November 1987

[PeGu99] Charles E. Perkins and Erik Guttman. RFC2610 - DHCP Options for Service Location Protocol, Proposed Internet Standard, June 1999

[Schn00] Bruce Schneier; Secrets and lies: digital security in a networked world, John Wiley and Sons, 2000, ISBN 0-471-25311-1

[ThNa98] S. Thomson, T. Narten; RFC 2462 - IPv6 Stateless Address Autoconfiguration, Proposed Internet Standard, December 1998

[VeGu97] John Veizades, Erik Guttman, Charles E. Perkins, Scott Kaplan, RFC2165 - Service Location Protocol, Proposed Internet Standard, June 1997

[Weis91] Mark Weiser, The Computer for the Twenty-First Century, Scientific American, p.94-100, September 1991